UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/645,819	08/20/2003	Robin J. Green	SONYP024	8040

25920 7590 02/22/2006

MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER	
REPKO, JASON MICHAEL	
ART UNIT	PAPER NUMBER
2671	

DATE MAILED: 02/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/645,819	**Applicant(s)** GREEN ET AL.
	Examiner Jason M. Repko	**Art Unit** 2671

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 December 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Drawings

1. The replacement drawings filed on 12/29/2005 have been accepted.

Response to Arguments

2. Applicant's arguments filed 12/29/2005 have been fully considered but they are not persuasive.

3. With regard to the argument "neither Sloan nor Purcell teach determining an approximation of a transfer function component using at least one basis function, the approximation corresponding to a center point of a texel..." as in the amended claims, Sloan et al discloses sampling texels (*5th paragraph of section 6.2: "The basis function textures are also supersampled and decimated in the same way as a preprocess."*) using the sampled texture to determine the lighting characteristics (*2nd paragraph of section 6.2: " For efficiency, we precompute textures for the basis functions weighted by differential solid angle...each evaluated over the cube map parameterization for s. The resulting integral then becomes a simple dot product of the captured samples of $L_P(s)$ with the textures $B^m_l(s)$."*). Sloan et al defines $L_P(s)$ as a parameterization of the incident lighting in the first paragraph of section 4, and Sloan et al uses the language "light function" and "transfer function" in referring to L_p and M_p, respectively, in the first paragraph of section 4.1. Sloan et al does not expressly disclose a sampling location. However, U.S. Patent Application Publication No. 2001/0028352 to Neagle et al teaches sampling the center of a pixel. As shown in the prior art rejection that follows, it would have been obvious to one of ordinary skill in the art to combine Sloan et al and Neagle et al.

4. With regard to the arguments "Purcell requires that the scene geometry is stored as triangles within a data structure" and "Purcell may not be efficient for dynamic scenes" in contrast to the claimed invention for "presenting lighting characteristics associated with a display object," Sloan et al discloses in the first paragraph of section 5: "As a preprocess, we perform a global illumination simulation over an object ***O*** using the SH basis over the infinite sphere as emitters," and Sloan et al references a ray-tracing algorithm ([7] Cook et al, "Distributed Ray Tracing"). The ray-tracing step in Sloan et al requires access to the scene's geometry stored as polygonal data in the same manner as the claimed invention (i.e. claims 6 and 7 of the instant application: "reading polygon data associated with the path [of a ray]."). Yet, as the reference indicates, Sloan et al is able to provide "real-time rendering in dynamic, low-frequency lighting environments," because the transfer functions are applied to the incident lighting to provide the lighting without having to access the scene's geometry once the data from the global illumination simulation is available (figure 2 and section 6). Sloan et al further suggests that the straightforward implementation could be accelerated. Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to map the computations required by rendering algorithm for dynamic, low-frequency lighting environments disclosed by Sloan et al to a stream processor as taught by Purcell et al to gain the advantage of the efficient implementation and parallelism of the stream processing embodiment to achieve better performance as shown in the rejection of claims 1-15, 17-24, 26, 28-31, 33, 34, and 36-38.

5. With regard to the statement "as acknowledged by the Examiner, Sloan is silent as to ray tracing," it was stated in the previous office action Sloan teaches "a ray tracing algorithm executing that includes generating a ray associated with a point on the display object" (see first

sentence of paragraph 8), but does not teach ray tracing on a stream processor (see last sentence of paragraph 8). Nowhere in the previous office action does the Examiner state anything to contradict these statements or indicate the Examiner acknowledges the Sloan et al is silent as to ray tracing. As shown in the preceding paragraph, Purcell et al teaches ray tracing on a stream processor and it would have been obvious to modify the Sloan et al system.

6. With regard to the statement "as acknowledged by the Examiner, Sloan et al does teach multiply and add operations," Sloan et al does teach the multiply and add operations (*stated in the 3rd paragraph of section 6: "For diffuse surfaces, a simple implementation of step 3 is to store the transfer vector per vertex and perform the dot product in a vertex shader."*); however, Sloan et al does not teach multiply and add operations in a ray tracer mapped to a stream processor. Purcell et al teaches ray tracing on a stream processor and it would have been obvious to modify the Sloan et al system.

7. With regard to the limitation added to claim 24, Sloan et al teaches "the multiply and add operations are performed by a processor without calculating the lighting function at triangle corners" (*3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader... Our pixel shader needs 8 instructions to perform the dot-product and stores LP's coefficients in constant registers."*). Purcell et al teaches ray tracing on a stream processor and it would have been obvious to modify the Sloan et al system.

8. With regard to the argument the feature of "applying a ray tracing algorithm through a stream processor without accessing pre-calculated geometry associated with the object" is patentable over the cited combination, the Sloan et al teaches using the dot product with texture

maps to determine lighting characteristics (*2^{nd} paragraph of section 6.2: " For efficiency, we precompute textures for the basis functions weighted by differential solid angle…each evaluated over the cube map parameterization for s. The resulting integral then becomes a simple dot product of the captured samples of $L_P(s)$ with the textures $B^m{}_l(s)$. "; 3^{rd} paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader.* ") As previously shown, Purcell et al teaches a stream processor that can be used to accelerate computations of the Sloan et al system, as the dot product can be performed in parallel according to the teachings of Purcell et al.

9. With regard to the amendment adding the limitation "determining a lighting characteristic by computing the lighting function at corners of a triangle," although Sloan et al does teach using the "corners of a triangle" in the second paragraph of section 6 (*"For diffuse surfaces, a simple implementation of step 3 is to store the transfer vector per vertex and perform the dot product in a vertex shader.* "); however, the alternate texel based implementation in the third paragraph of section 6 (*"The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader*") reads on the claim language added by amendment. Therefore, the limitation "calculating a value representing a lighting characteristic for each of the texels without calculating a lighting function at triangle corners," does not patentably distinguish the claimed invention from the Sloan et al reference.

Claim Rejections - 35 USC § 112

10. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

11. Claim 6-11 and 17-22 rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

12. With regard to the language "generating a ray from a point on the object without accessing pre-calculated geometry associated with the object" in claims 6 and 17, the step of "generating a ray associated with a point on the display object" appears in paragraphs 12, 13, 14, 15, 16, 53 (line 6-7, p. 16) and 67 in describing a ray tracing algorithm, but these paragraphs do not disclose how to generate a ray from a point on the object without accessing the pre-computed geometry with the object. With respect to "determining secondary illumination features," the Applicant discloses that the ray tracing calculations are first computed to determine "the geometry associated with the object and the object's environment" (lines 10-11, p. 30) and secondary lighting contribution is determined using the data resulting from the ray tracing, but does not describe "generating a ray from a point on the object without accessing pre-calculated geometry associated with the object." Furthermore, paragraphs 72-76, Fig. 17, and Fig. 18 do not convey to one skilled in the relevant art how generating rays can be accomplished in this manner.

Claim Rejections - 35 USC § 103

13. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

14. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

15. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

16. **Claims 1-5, 12-15 and 28-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peter-Pike Sloan, Jan Kautz, John Snyder, "Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments," July 2002, ACM Transactions on Graphics (TOG), v. 21 n. 3 (herein referred to as "Sloan et al") in view of Timothy J. Purcell, Ian Buck, William R. Mark, Pat Hanrahan, "Ray tracing on programmable graphics hardware," July 2002, In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques (herein referred to as "Purcell et al") in further view of U.S. Patent Application Publication No. 2001/0028352 to Neagle et al (herein referred to as "Neagle et al").**

17. With regard to **claim 1**, Sloan et al teaches "a method for presenting lighting characteristics associated with a display object in real-time, comprising: executing a ray tracing algorithm that includes generating a ray associated with a point on the display object" (*5th paragraph of section 5: "In the first pass, for each p $\in$ O, we cast shadow rays in the hemisphere about p's normal Np, using the hierarchy to cull directions outside the hemisphere."*). Sloan et al teaches "determining an approximation of a transfer function component using at least one basis function" in the equations given in the 6th and 7th paragraphs of section 5. Sloan et al does not use this explicit language; however one of ordinary skill in the art would recognize this feature from the description of the terms of the equation in the 2nd paragraph of section 5 and the 3rd paragraph of section 4:

> *A transfer matrix $(\mathcal{M}_p)_{ij}$ is useful for glossy surfaces and represents a linear transformation on the lighting vector which produces projection coefficients for an entire spherical function of transferred radiance $L'_p(s)$ rather than a scalar...Components of $(\mathcal{M}_p)_{ij}$ represent the linear influence of the j-th lighting coefficient of incident radiance $(L_p)_j$ to the i-th coefficient of transferred radiance$(L'_p)_i$.*

18. Sloan et al does not teach ray tracing on stream processor. Purcell et al teaches "executing a ray tracing algorithm through a stream processor" (*Figure 2, page 705*).

19. With regard to **claim 28**, Sloan et al teaches "calculating a lighting function for an object to be rendered using a basis function, comprising: calculating a transfer function approximation of the lighting function," (*3rd paragraph of section 4: "A transfer matrix $(M_p)_{ij}$ is useful for glossy surfaces and represents a linear transformation on the lighting vector which produces projection coefficients for an entire spherical function of transferred radiance $L'_p(s)$ rather than*

a scale.") "the lighting function being sampled at a texel" (*3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader."; 5th paragraph of section 6.2: "The basis function textures are also supersampled and decimated in the same way as a preprocess."*). Furthermore, Sloan et al states in the 4th paragraph of section 1:

> *The object's shaded "response" to its environment can be viewed as a transfer function, mapping incoming to outgoing radiance, which in this case simply performs a cosine-weighted integral.*

20. Sloan et al does not teach using a stream processor to compute the rays, which determines the transfer function (*1st paragraph of section 5: "As a preprocess, we perform a global illumination simulation over an object **O** using the SH basis over the infinite sphere as emitters."*). Purcell et al teaches "executing a ray tracing algorithm through a stream processor" (*Figure 2, page 705*).

21. With regard to **claims 1 and 28**, Sloan et al states that the "light gathering solution technique is a straightforward adaptation of existing approaches [7][25] and could be accelerated in many ways," noting that [7] is a ray-tracing algorithm. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to accelerate the computation of radiance transfer as taught by Sloan et al, by using a stream processor to trace rays as taught by Purcell et al. The motivation for doing so would have been to achieve better performance tracing rays as stated by Purcell et al in section 2.3, which would be advantageous for interactive systems.

22. With regard to **claims 1 and 28**, Sloan et al discloses sampling a texel associated with a corresponding point on an object (*1st paragraph of section 6: " We now have a model **O** capturing radiance transfer at many points p over its surface, represented as vectors or matrices."*; *3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader."; 5th paragraph of section 6.2: "The basis function textures are also supersampled and decimated in the same way as a preprocess."*), but does not disclose a sampling location within the texel. Neagle et al discloses sampling at the "center point of a pixel" (*paragraph 256: "In one embodiment, the graphics system ensures that one of the rendered samples lies in the center of the bin or pixel area."*).

23. With regard to **claims 1 and 28**, at the time of the invention, it would have been obvious to a person of ordinary skill in the art to use the sampling scheme disclosed by Neagle et al to sample the texels in the system and method disclosed by Sloan et al. The motivation for doing so would have been to sample at a location that is more representative of the given texel than any neighboring texel, an advantage well-known in the art. Therefore, it would have been obvious to combine Sloan et al with Neagle et al to obtain the invention as specified in **claims 1 and 28.**

24. **Claim 2** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches "determining whether the ray is within a view plane of a light source" (*5th paragraph of section 5: "In the first pass, for each p $\in$ O, we cast shadow rays in the hemisphere about p's normal Np... We tag each direction s_d with an occlusion bit, 1 – $V_p(s_d)$, indicating whether s_d is in the hemisphere and intersects O again (i.e., is self-shadowed by O)."*). Sloan et al does not use this explicit language in the 5th paragraph of section 5; however, one of

ordinary skill in the art would recognize that a ray occluded by an object is not in the view plane of the light.

25. **Claim 3** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches "if the ray is within the view plane of the light source, then the method includes, determining a direct illumination component of the lighting characteristic" (*5th paragraph of section 5: "completely unoccluded bins/samples receive only direct light from the environment."*).

26. **Claim 4** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches "if the ray is not within the view plane of the light source, then the method includes, determining a self interreflection component of the lighting characteristic" (*5th paragraph of section 5: "Self-occluded directions and bins are tagged so that we can perform further interreflection passes on them"*).

27. **Claim 5** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches "repeating the determining of an approximation of a transfer function component for a series of basis functions without accessing a pre-calculated geometry associated with the object"(*2nd paragraph of section 4: "In other words, each component of $(M_p)_i$ represents the linear influence that a lighting basis function $(L_p)_i$ has on shading at p."; 2nd paragraph of section 6.2: " For efficiency, we precompute textures for the basis functions weighted by differential solid angle...each evaluated over the cube map parameterization for s. The resulting integral then becomes a simple dot product of the captured samples of $L_P(s)$ with the textures $B^m{}_l(s)$."; 3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader."*); and rendering the

display object using the approximation of the transfer function component for the series of basis functions" (*1st paragraph of section 6: "Rendering [model] **O** requires the following steps at run-time:...compute incident lighting $\{L_{Pi}\}$ at one or more sample points P_i near **O** in terms of the SH basis."*).

28. **Claims 12-15** are rejected with the rationale of claims 1-4, respectively. Claims 12-15 are similar in scope to claims 1-4 recited as a computer readable medium having program instructions (*shown in section 9 of Sloan et al*).

29. **Claim 29** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches a "transfer function approximation is associated with the basis function that characterizes a global illumination associated with the object" (*2nd paragraph of section 4: "A transfer vector $(M_p)_i$ is useful for diffuse surfaces and represents a linear transformation on the lighting vector producing scalar exit radiance...each component of $(M_p)_i$ represents the linear influence that a lighting basis function $(L_p)_i$ has on shading at p."*).

30. **Claim 30** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches a "transfer function approximation is a set of coefficients configured to describe a surface reflectance" (*4th paragraph of section 4: "Components of $(\mathcal{M}_p)_{ij}$ represent the linear influence of the j-th lighting coefficient of incident radiance $(L_p)_j$ to the i-th coefficient of transferred radiance(L'_p)"*).

31. **Claim 31** is met by the combination of Sloan et al, Purcell et al and Neagle et al, wherein Sloan et al teaches rendering an object (*1st paragraph of section 6: "We now have a model **O** capturing radiance transfer at many points p over its surface, represented as vectors or matrices. Rendering O requires the following steps at run-time:..."*).

32. Claims 6-11, 17-24, 26, 33, 34, and 36-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al in view of Purcell et al.

33. With regard to **claim 6**, Sloan et al teaches "a method for determining secondary illumination features for an object to be displayed, comprising: calculating an approximation to a transfer function associated with at least one basis function" in the equations given in the 6th and 7th paragraphs of section 5 (as shown with regard to claim 1); "wherein the approximation to the transfer function represents a component of the secondary illumination features" (*4th paragraph of section 5: "Components of* $(\mathcal{M}_p)_{ij}$ *represent the linear influence of the j-th lighting coefficient of incident radiance* $(L_p)_j$ *to the i-th coefficient of transferred radiance*$(L'_p)_i$*.").* Sloan et al teaches generating a ray without accessing pre-calculated geometry associated with the object (*5th paragraph of section 5: "In the first pass, for each* $p \in \mathbf{O}$*, we cast shadow rays in the hemisphere about p's normal Np..."*), and "determining if a ray intersects a surface" (*5th paragraph of section 5*: "*We tag each direction* s_d *with an occlusion bit,* $1 - V_p(s_d)$*, indicating whether* s_d *is in the hemisphere and intersects* ***O*** *again (i.e., is self-shadowed by* ***O****)."*); however, Sloan et al does not teach ray tracing on a stream processor.

34. Purcell et al teaches "providing a stream processor capable of identifying a path associated with a ray" (*page 705, 1st paragraph of section 2.3: "In order to guide the mapping of new applications to graphics architectures, we propose that we view next generation graphics hardware as a streaming processor."; 4th paragraph of section 3, titled "Streaming Ray Tracing": "The traversal kernel steps rays through the grid until the ray encounters a voxel containing triangles"*); "generating a ray from a point on the object" (*1st paragraph of section 3.1.2: "The traversal stage searches for voxels containing triangles...The second part steps*

along the ray enumerating those voxels pierced by the ray. "); "determining if the path of the ray intersects a surface" (*4th paragraph of section 3, titled "Streaming Ray Tracing": "The intersection kernel is responsible for testing ray intersections with all the triangles contained in the voxel.* ").

35. With regard to **claim 23**, Sloan et al teaches "a method for calculating an approximation to a transfer function defined by at least one basis function for rendering shading characteristics of an object in real time, comprising: identifying a point on the object; calculating a lighting function for the point" (*1st paragraph of section 5: "As a preprocess, we perform a global illumination simulation over an object **O** using the SH basis over the infinite sphere as emitters."; 6th paragraph of section 5: "For diffuse surfaces, at each point p ∊ **O** we further compute the transfer vector by SH-projecting M_p [transfer function] from [equation] (10).* "). Sloan et al teaches "determining a direct illumination transfer function for the point" by means of a ray tracer (*1st paragraph of section 5: "As a preprocess, we perform a global illumination simulation over an object **O** using the SH basis over the infinite sphere as emitters."; 3rd paragraph of section 5: "An initial pass simulates direct shadows from paths leaving L and reaching sample point p ∊ **O**.* "). Sloan et al teaches applying a ray tracing algorithm without accessing pre-calculated geometry associated with the object (*4th paragraph of section 4: "...$L_p(s)$ represents incident lighting assuming **O** was removed from the scene."; 1st paragraph of section 6.2: "**O** itself should be removed from these renderings.* "). Sloan et al does not teach ray tracing on a stream processor in real-time.

36. Purcell et al teaches "applying a ray tracing algorithm through a stream processor" (*Figure 2, page 705*). Purcell et al teaches "determining a secondary lighting contribution in real

time through a series of multiply and add operations applied to data resulting from the ray tracing algorithm" (*4th paragraph of section 3: "Additionally, the shading kernel may generate shadow or secondary rays; in this case, these new rays are passed back to the traversal stage."*). With regard to the multiply and add operations for secondary lighting for secondary illumination, the triangle intersection stage (as shown in Figure 2 on page 705 of Purcell et al) occurs after the traversal stage. Figure 5 (Code for ray-triangle intersection) shows the dot product operation for two vectors on lines 12, 14, and 15. Furthermore, Purcell et al refers to the method as "real-time ray tracing" in the first sentence of the 2nd paragraph of section 1.

37. With regard to **claims 6 and 23**, Sloan et al states that the "light gathering solution technique is a straightforward adaptation of existing approaches [7][25] and could be accelerated in many ways," noting that [7] is a ray-tracing algorithm. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to accelerate the computation of radiance transfer as taught by Sloan et al, by using a stream processor to trace rays as taught by Purcell et al. The motivation for doing so would have been to achieve better performance tracing rays as stated by Purcell et al in section 2.3, which would be advantageous for interactive systems. Therefore, it would have been obvious to combine Sloan et al with Purcell et al to obtain the invention as specified in **claims 6 and 23**.

38. **Claim 7** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches a "stream processor capable of identifying a path associated with a ray includes, reading data associated with the ray" (*4th paragraph of section 3: "The traversal kernel reads the stream of rays produced by the eye ray generator."; 1st paragraph of section 3.1.2: "The second part steps along the ray enumerating those voxels pierced by the ray")*; and "reading polygon data

associated with the path" (*4th paragraph of section 3: "The traversal kernel steps rays through the grid until the ray encounters a voxel containing triangles. The ray and voxel address are output and passed to the intersection kernel."; Figure 4).*

39. **Claim 8** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches an "operation of generating a ray from a point on the object includes, determining a voxel traversed by a ray segment" (*1st paragraph of section 3.1.2: "The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray."*).

40. **Claim 9** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches "if the path of the ray does not intersect the surface, then the method includes, determining a next voxel traversed by a next ray segment" (*3rd paragraph of section 3: "If no hit occurs, the ray is passed back to the traversal kernel and the search for voxels containing triangles continues."*).

41. **Claim 10** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches "if the ray intersects a surface, then the method includes, recording data associated with the location of the surface intersection" (*4th paragraph of section 3: "If ray-triangle intersection (hit) occurs in that voxel, the ray and the triangle that is hit is output for shading."*); and "generating a next ray" (*3rd paragraph of section 3.1.4: "The shading kernel optionally generates shadow, reflection, refraction, or randomly generated rays."*); and "if the ray does not intersect a surface, then the method includes, reading data associated with a next voxel" (*3rd paragraph of section 3: "If no hit occurs, the ray is passed back to the traversal kernel and the search for voxels containing triangles continues."; 2nd paragraph of section 3.1.2: "After each*

step, the kernel queries the grid data structure which is stored as a 3D texture."); "advancing the ray through the next voxel" (*1st paragraph of section 3.1.2: "The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray."*); "and repeating if the path of the ray intersects the surface" (*1st paragraph of section 3.1.2: "The traversal stage searches for voxels containing triangles...The second part steps along the ray enumerating those voxels pierced by the ray."*).

42. **Claim 11** is met by the combination of Sloan et al and Purcell et al, wherein Purcell et al teaches, "determining if the surface intersection is a closest surface intersection" (*Figure 3 (c)).*

43. **Claims 17-22** are rejected with the rationale of claims 6-11 respectively. Claims 17-22 are similar in scope to claims 6-11 recited as a computer readable medium with having program instructions.

44. With regard to **claim 24,** Sloan et al further teaches "the multiply and add operations are performed by a processor without calculating the lighting function at triangle corners" (*3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader...Our pixel shader needs 8 instructions to perform the dot-product and stores LP's coefficients in constant registers."*). Sloan et al does not teach a stream processor. Purcell et al teaches a stream processor (*1st paragraph of section 2.3: "In order to guide the mapping of new applications to graphics architectures, we propose that we view next-generation graphics hardware as a streaming processor."*).

45. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to further modify the combination of Sloan et al and Purcell et al to incorporate computing the dot product in the stream processor. The motivation for doing so would have been

to achieve better performance resulting from parallel computations, as suggested by Purcell et al in the third paragraph of section 2.3. Therefore, it would have been obvious to further modify Sloan et al with Purcell et al to obtain the invention specified in **claim 24.**

46. **Claim 26** is met by the combination of Sloan et al and Purcell et al, wherein Sloan et al teaches "repeating the identifying and the calculating for multiple points on the object" (*2nd paragraph of section 5: "An initial pass simulates direct shadows from paths leaving L and reaching sample points p $\in$ O...In each pass, energy is gathered to every sample point p."*).

47. **Claims 33, 34, and 36-38** are rejected with the rationale of claims 23, 26, and 28-30 respectively. Claims 33, 34, and 36-38 are similar in scope to claims 23, 26, and 28-30 recited as a computer readable medium with having program instructions.

48. **Claims 16 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al in view of Purcell et al and in further view of Robert L. Cook, "Stochastic sampling in computer graphics," Jan. 1986, ACM Transactions on Graphics (TOG), v.5 n.1, p.51-72 (herein referred to as "Cook.")**

49. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of parent claim 12. With regard to claim 16, the Sloan et al and Purcell et al combination is silent as to biased or unbiased approximators. Cook et al teaches "applying one of a biased" approximator (*1st paragraph of section 5.2: "Sometimes we need to weight the samples...A better approach is importance sampling, in which the sample points are distributed so that the chance of a location being sampled is proportional to the value of the filter at that location."; 3rd paragraph of section 5.2: "For example, for the reflection ray, we create a lookup table based on the specular*

reflection function.") and "unbiased approximator" (*1st paragraph of section 5.1: "One way to distribute the rays in the additional dimensions is with uncorrelated random values.").*

50. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of parent claim 23. With regard to claim 25, the Sloan et al and Purcell et al combination is silent as to biased or unbiased approximators. Cook et al teaches "applying one of a biased approximator" in the 1st paragraph of section 5.2 as shown in the paragraph above.

51. Cook et al, Sloan et al and Purcell et al are analogous art because they are from the same problem solving area: computing illumination on graphics hardware. At the time of the invention it would have been obvious to a person of ordinary skill in the art to apply one of an unbiased approximator and the biased approximator as taught by Cook et al in the ray tracer taught by the Purcell et al and Sloan et al combination. The motivation for doing so would be to provide a pattern to distribute the rays, or in some cases, such as determining direct lighting, to use a biased approximator to direct where the rays will go to avoid unproductive computations (*Cook et al in the 1st paragraph of section 5.2 states that importance sampling "puts the samples where they will do the most good.").* Therefore, it would have been obvious to combine Sloan et al and Purcell et al with Cook to obtain the invention as specified in claims 16 and 25.

52. Claims 27 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al in view of Purcell et al and in further view of U.S. Patent No. 6,268,860 to Bonello (herein referred to as "Bonello.")

53. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of claim 31. With regard to claim 27, Sloan et al does not disclose delayed evaluation based on frames. Bonello discloses performing a calculation "for a portion of the multiple points during a

first frame of image data, and performing the calculation for a remainder of the multiple points during a next frame of image data" (*lines 41-47 of column 8: " In this variation, the invention makes use of the knowledge that the illumination situation rarely changes between images (frames) that are adjacent in time, so it suffices to calculate precisely the relevance of the individual light sources for only every third frame, for example, and retain it in the display of the frames located between them in time."*).

54. Sloan et al, Purcell et al and Bonello are analogous art because they the same problem solving area: improving the efficiency of illumination calculations for image generation. At the time of the invention it would have been obvious to a person of ordinary skill in the art to defer running illumination calculation on a portion of the data as taught by Bonello in the image generation system as taught by the Sloan et al and Purcell et al combination. The motivation for doing so would have been to reduce the amount of computation for displaying computer models for each frame as stated by Bonello in lines 31-32 of column 8, which would improve the performance of an interactive system.

55. Claim 35 is rejected with the rationale of claim 27. Claim 35 is claim 27 recited as a computer readable medium.

56. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sloan et al in view of Purcell et al and in further view of Tomas Möller, Eric Haines, "Real-Time Rendering," 1999, A.K. Peters, p. 68 (herein referred to as "Möller et al.")

57. As previously stated, the Sloan et al and Purcell et al combination meets the limitations of claim 31. With regard to claim 32, Sloan et al does not disclose linear interpolation. Möller et al teaches, "linearly interpolating a color of the object across a polygon" (*2nd paragraph of p. 68:*

"In Gouraud shading [138], the lighting at each vertex of a triangle is determined, and these lighting samples (i.e., computed colors) are interpolated over the surface of the triangle.").

58. Sloan et al, Purcell et al and Möller et al are analogous art because they are from the same problem solving area: real-time rendering. Möller et al states in the 3rd paragraph on page 68, "Most graphics hardware implement Gouraud shading because of its speed and much improved quality." At the time of the invention it would have been obvious to a person of ordinary skill in the art to incorporate interpolation as taught by Möller et al in the color computations of the combination of Sloan et al and Purcell et al. The motivation for doing so would have been to give a "smooth look to curved surfaces" (3rd paragraph of page 68, Möller et al) without having to compute the color attribute at each point on the surface of the object.

59. Claims 39-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,639,595 to Drebin et al (herein referred to as "Drebin et al") in view of Sloan et al and in further view of Purcell et al.

60. With regard to **claim 39**, Drebin et al discloses "A computing device, comprising: a graphics processing unit (GPU) capable of determining lighting characteristics for an object in real time" (*lines 58-60 of column 4: "In this example, system 50 is capable of processing, interactively in real time, a digital representation or model of a three-dimensional world."; lines 9-13 of column 9: "As discussed above, transform unit 300 in the example embodiment performs lighting in addition to geometric transformations, clipping, culling and other functions. In the example embodiment, transform unit 300 supports lighting in hardware as a per-vertex calculation."*). Drebin et al does not disclose a basis function or a stream processor.

61. Sloan et al discloses “determining lighting characteristics for an object in real time without accessing pre-calculated geometry associated with the object” (*3rd paragraph of section 6: “The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader.”; 2nd paragraph of section 6.2: " For efficiency, we precompute textures for the basis functions weighted by differential solid angle...each evaluated over the cube map parameterization for s. The resulting integral then becomes a simple dot product of the captured samples of* $L_P(s)$ *with the textures* $B^m_l(s)$*."*) the lighting characteristics defined through a basis function" (*1st paragraph of section 4: "...we first parameterize incident lighting at points* $p \in O$*, denoted* $L_p(s)$*, using the [Spherical Harmonic] basis."*). Purcell et al teaches, “a stream processor configured to split a stream of data associated with the lighting characteristics into multiple simultaneous operations” (*3rd paragraph of section 2.3: “First, since each stream element ‘s computation is independent from any other, designers can add additional pipelines that process elements of the stream in parallel.”*).

62. At the time of the invention it would have been obvious to a person of ordinary skill in the art to define the lighting characteristics of Drebin et al through a basis function as taught by Sloan et al. The motivation for doing so would have been provide a framework to represent complex light transport efficiently as stated by Sloan et al in the 3rd paragraph of section 1. Furthermore, at the time of the invention it would have been obvious to a person of ordinary skill in the art to include the stream processor taught by Purcell et al on the GPU in the graphics-processing unit disclosed by Drebin et al to carry out the computations needed by the method taught by Sloan et al. The motivation to combine the Drebin et al and Sloan et al combination

with Purcell et al would have been to efficiently implement ray tracing as stated by Purcell et al in the 3rd paragraph of section 2.3.

63. **Claim 40** is met by the combination of Drebin et al, Sloan et al and Purcell et al, wherein Drebin et al discloses “a display screen in communication with the GPU, the display screen configured to present image data representing the object” (*lines 3-5 of column 6: “The graphics and audio processor 114 processes these commands to generate interesting visual images on display 59... "*).

64. **Claim 41** is met by the combination of Drebin et al, Sloan et al and Purcell et al, wherein Drebin et al discloses a computing device that is a “video game console" (*lines 5-8 of column 5: “To play a video game or other application using system 50, the user first connects a main unit 54 to his or her color television set 56 or other display device by connecting a cable 58 between the two. "*).

65. **Claim 42** is met by the combination of Drebin et al, Sloan et al and Purcell et al, wherein Purcell et al teaches a "stream processor is a programmable hardware unit capable of executing code that is replicated multiple times" (*paragraph 2 of section 2.3: "The system executes a program or kernel on each element of the input stream...In this sense, a programmable graphics processor executes a vertex program on a stream of vertices... "; 3rd paragraph of section 2.3: "First, since each stream element 's computation is independent from any other, designers can add additional pipelines that process elements of the stream in parallel. "*).

66. **Claim 43** is met by the combination of Drebin et al, Purcell et al and Sloan et al, wherein Purcell et al teaches "code that is replicated multiple times is configured to process one of a ray tracing algorithm" (*Figure 2: A streaming ray tracer; 5th paragraph of section 6: “We implement*

ray tracing kernels as fragment programs.") and "multiply and add operations in the stream processor" (*Figure 5: code for ray tracing triangles.*). Purcell et al does not use the explicit language "multiply and add operations", but one of ordinary skill in the art would recognize that this feature is inherent from lines 12, 14, and 15 in the code presented in Figure 5, where the dot product of two vectors is computed.

67. **Claim 45** is met by the combination of Drebin et al, Sloan et al and Purcell et al, wherein Drebin et al discloses "the GPU is further configured to render the object through a process involving linear interpolation, such that the lighting characteristics are applied to the rendered object" (*lines 11-16 of column 9: "In the example embodiment, transform unit 300 supports lighting in hardware as a per-vertex calculation. This means that a color (RGB) value can be computed for every lit vertex, and that these colors can be linearly interpolated over the surface of each lit triangle."*).

68. With regard to **claim 44**, Sloan et al further teaches, "the ray tracing algorithm determines a direct illumination lighting characteristic in real time for multiple points on an object" (*3rd paragraph of section 5: "An initial pass simulates direct shadows from paths leaving L and reaching sample points p $\in$ **O**"*) and "the multiply an add operation determine a secondary lighting characteristic in real time" (*1st paragraph of section 6: "We now have a model **O** capturing radiance transfer at many points p over its surface, represented as vectors or matrices."; 3rd paragraph of section 6: "The transfer vectors can also be stored in texture maps rather than per-vertex and evaluated using a pixel shader."; 1st paragraph of section 9: "We achieve real-time performance for all models except the transfer matrix ones (teapot, buddha, glider).*).

69. At the time of the invention, it would have been obvious to a person of ordinary skill in the art to further modify the combination of Sloan et al and Purcell et al to incorporate computing the dot product in the stream processor. The motivation for doing so would have been to achieve better performance resulting from parallel computations, as suggested by Purcell et al in the third paragraph of section 2.3. Therefore, it would have been obvious to further modify Drebin et al, Purcell et al and Sloan et al to obtain the invention specified in **claim 44.**

70. **Claim 46** is met by the combination of Drebin et al, Purcell et al and Sloan et al, wherein Sloan et al teaches the basis function is a spherical basis function (*1st paragraph of section 4: "...we first parameterize incident lighting at points p* $\in$ ***O****, denoted* $L_p(s)$*, using the [spherical harmonic] basis."*).

71. At the time the invention was made, it would have been an obvious matter of design choice to a person of ordinary skill in the art to use "one of a wavelet and a spherical basis function" because Applicant has not disclosed that a wavelet basis provides an advantage. One of ordinary skill in the art, furthermore, would have expected a basis function, and applicant's invention, to perform equally well as either a spherical harmonic basis taught by Sloan et al or the wavelet basis because both are suitable basis functions, as described in the specification of the instant application (*paragraph 48: "Additionally, the transfer and illumination functions may be expressed as the sum of any suitable basis functions. Exemplary basis functions include spherical harmonic basis functions, wavelet"*), and in Sloan et al (*paragraph 4 of section 9: "Because of its rotational invariance (equation (4)), we consider the SH basis especially useful for our low-frequency lighting application compared to alternatives like spherical wavelets [37]."*).

72. Therefore, it would have been prima facie obvious to modify the Drebin et al, Purcell et al, and Sloan et al combination to obtain the invention as specified in claim 46 because such a modification would have been considered a mere design consideration which fails to patentably distinguish over the prior art of the Drebin et al, Purcell et al, and Sloan et al combination.

Conclusion

73. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Nathan A. Carr, Jesse D. Hall and John C. Hart, "The Ray Engine," March 2002, Tech. Rep. UIUCDCS-R-2002-2269, Department of Computer Science, University of Illinois, teach a ray-triangle intersection computation using a pixel shader. John D. Owens, William J. Dally, Ujval J. Kapasi, Scott Rixner, Peter Mattson, Ben Mowery, “Polygon rendering on a stream architecture,” August 2000, Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, p.23-32, teaches polygon rendering on a stream architecture. Stephen H. Westin, James R. Arvo, Kenneth E. Torrance, "Predicting reflectance functions from complex surfaces," July 1992, Proceedings of the 19th annual conference on Computer graphics and interactive techniques, p.255-264 teaches using a spherical harmonic basis for lighting calculations.

74. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason M. Repko whose telephone number is 571-272-8624. The examiner can normally be reached on Monday through Friday 8:30 am -5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on 571-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JMR

U Chauhan
ULKA CHAUHAN
SUPERVISORY PATENT EXAMINER